I. REAL PARTY IN INTEREST 1

II. RELATED APPEALS AND INTERFERENCES 1

III. STATUS OF CLAIMS 2

IV. STATUS OF AMENDMENTS 2

V. SUMMARY OF CLAIMED SUBJECT MATTER 2

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL 3

VII. ARGUMENT 4

VIII. CLAIMS APPENDIX 28

IX. EVIDENCE APPENDIX 32

X. RELATED PROCEEDINGS APPENDIX 33

Docket No. JP920030233 (7161-128U) **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	Customer Number: 46320
Scott CARRIER	:	Confirmation Number: 9171
Application No.: 10/712,544	:	Group Art Unit: 2176
Filed: November 13, 2003	:	Examiner: M. Botts

For: LIGHTWEIGHT FORM PATTERN VALIDATION

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is submitted in support of the Notice of Appeal filed February 15, 2007, wherein Appellant appeals from the Examiner's rejection of claims 1-15.

I. REAL PARTY IN INTEREST

This application is assigned to IBM Corporation by assignment recorded on November 13, 2003, at Reel 014707, Frame 0146.

II. RELATED APPEALS AND INTERFERENCES

Appellant is unaware of any related appeals and interferences.

III. STATUS OF CLAIMS

Claims 1-15 are pending and three-times rejected in this Application. It is from the multiple rejections of claims 1-15 that this Appeal is taken.

IV. STATUS OF AMENDMENTS

The claims have not been amended subsequent to the imposition of the Third Office Action dated November 13, 2006 (hereinafter the Third Office Action).

V. SUMMARY OF CLAIMED SUBJECT MATTER

Referring to Figures 1 and 2 and also to independent claim 1, a lightweight pattern validation system for a client device 130 receiving markup 200 defining a form 160 is disclosed (paragraph [0017] of Appellant's disclosure). The system includes a validation processor 150 and a validation script library 140 (paragraph [0018]). The validation processor 140 is separate from the markup 200 and configured with a prototype interface for receiving both a field validation pattern 170 and also form based input 170 to be validated against the field validation pattern 170 (paragraph [0018]). The validation script library 140 is within the client device 130 and packages the validation processor 150 (paragraph [0018]). The form 160 has at least one form based input field programmed for validation using the validation processor 150 (paragraph [0019]).

Referring to Figure 3 and also to independent claims 6 and 11, a pattern validation method is disclosed. In block 310, a value for a form based input field is retrieved from a form defined in markup rendered in a content browser (paragraph [0023]). In block 330, the retrieved value along with a validation pattern for the form based input field is passed to a validation process disposed within a lightweight validation library separate from and coupled to the rendered markup (paragraph [0023]). In block 350, the retrieved value is validated according to the validation pattern in the content browser (paragraph [0024]).

Referring to Figure 2 and also to independent claim 10, a pattern validation method is disclosed. A pattern validation routine 220 is defined to validate form based input provided through a prototype interface to the routine based upon a validation pattern also provided through the prototype interface (paragraph [0022]). The pattern validation routine 220 is packaged into a lightweight validation script library 210 (paragraph [0021]). The lightweight validation script library 210 is referenced in markup 200 disposed within a content server configured to distribute the markup 200 to requesting clients (paragraph [0021]). At least one form based input field 250 in the markup 200 is defined (paragraph [0020]). A validation pattern for each of the at least one form based input fields 250 is defined (paragraph [0024]). For each form based input field 250 and defined validation pattern, a function call 240 to the pattern validation routine is disposed in the lightweight script library 210 (paragraph [0022]).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1-15 were rejected under the first paragraph of 35 U.S.C. § 112;
2. Claim 15 was again rejected under the first paragraph of 35 U.S.C. § 112;
3. Claims 1-15 were rejected under the second paragraph of 35 U.S.C. § 112;
4. Claims 1-15 were again rejected under the second paragraph of 35 U.S.C. § 112; and
5. Claims 1-15 were rejected under 35 U.S.C. § 102 for anticipation based upon Scholz et al., U.S. Patent Publication No. 2003/0078949 (hereinafter Scholz).

VII. ARGUMENT

THE REJECTIONS OF CLAIMS 1-15 UNDER THE FIRST AND SECOND PARAGRAPHS 35 U.S.C. § 112

For convenience of the Honorable Board in addressing the rejections, each of claims 1-15 stand or fall alone.

In the First and Second Office Actions, the Examiner did not reject claims 1-15 under the first and second paragraphs of 35 U.S.C. § 112. Instead, the Examiner merely objected to certain claim terms as "requiring interpretation or definition." Appellant responded by noting that interpreting claim language (i.e., claim construction) is a normal and expected part of examination a patent application. However, the fact that the Examiner believes certain claimed terms may require claim construction is not the basis for an objection. Moreover, although the Examiner stated that "[a]ppropriate correction is required," Appellant is unclear as to both the legal basis for requiring correction and what that correction might be.

In the Third Office Action, the Examiner's objections to certain claim terms morphed into rejections under the first and second paragraphs of 35 U.S.C. § 112 that are both legally and factually deficient. For ease of addressing these rejections, Appellant has reproduced each of the claimed terms at issue and will address both 35 U.S.C. § 112 issues as to these terms.

Field Validation Pattern

This term is found, for example, in claim 1, which recites "a prototype interface for receiving both a field validation pattern and also form based input to be validated against said field validation pattern." This phrase was found in claim 1, as originally presented, and this phrase alone is sufficiently enabling to one having ordinary skill in the art. As recited, the form based input is validated against the field validation pattern. Thus, the field validation pattern is a device for comparing/validating the input a field to pattern.

A further discussion of pattern validation is found in paragraph [0003] of Appellant's disclosure and reproduced below:

> As in the case of any application, regardless of its mode of distribution, oftentimes, validation will be required for each input field in a form defined within an interface in which the end user can provide ad hoc input. Examples of input fields which might require validation can include free-form text input fields in which the end user is free to provide any time of input able to provided within the field. As the application logic may expect input of a particular format or pattern, however, validation will be required. In this regard, pattern validation refers to the inspection of user input to ensure that the input conforms to a particular pattern. Examples include date formats, time formats, credit card number formats, address formats, telephone number formats and the like. (emphasis added)

Thus, not only is the term "field validation pattern" enabled, one having ordinary skill in the art would have no difficultly understanding the scope of this term, particularly when reasonably interpreted in light of the written description of the specification.

Markup

Claim 2, as originally presented, recited "markup defining a form." This language was amended, in part, and claim 1 now recites "[a] lightweight pattern validation system for a client device receiving markup defining a form." In the paragraph spanning pages 10 and 11 of the Third Office Action, the Examiner asserted the following:

> "Markup" was alternatively known to one of ordinary skill in the art at the time of the invention as: The collection of tags describing the specifications on an electronic document, as for formatting." See, "The American Heritage College Dictionary," fourth edition, Houghton Muffin, Co., 2002, definition of "markup." This definition is consistent with the Examiner's interpretation of the term "disposed in markup" as meaning use of a markup language, however, that interpretation was denied by Applicant. See, Remarks, page 8, noting that Applicant failed to state what the term was intended to mean.

At the outset, Appellant notes that Appellant objected to the interpretation of the term "'disposed in markup' as meaning use of a markup language." This issue of the definition of the phrase "disposed in markup" is separate from the issue of the definition of "markup." The term "markup" is commonly used shorthand for the term "markup language document," and the term "markup language document" is ubiquitous within Appellant's disclosure. Thus, one having ordinary skill in the art would have no difficultly understanding the scope of the term "markup," particularly when reasonably interpreted in light of the written description of the specification.

Disposed in said Markup

With regard to this phrase, the Examiner has argued the following in the paragraphs spanning pages 4 and 5 and also on page 8 of the Third Office Action:

> The term "disposed within markup" is not defined in the specification. It is the belief of the Examiner, based on a review of the claims and specification, that the Applicant intended to the term to mean that the invention used a markup computer language, and the term will be so read for the remainder of this Office Action.

On page 8 of the Second Amendment, Appellant argued the following:

> As another example, the Examiner asserted that the term "disposed within markup" means that "the invention used a markup computer language." How the Examiner arrived at this interpretation, however, is again unclear. Notwithstanding the Examiner's lack of analysis, the Examiner has improperly failed to recognize that the term "disposed within markup" establishes a relationship between a library reference and markup such that "a library reference ... [is] disposed within markup," as recited in claim 2. By merely asserting that the invention uses a markup computer language, the Examiner has ignored this claimed relationship between the library reference and markup.

Appellant continues to stand by this argument. Moreover, Appellant is unclear as to why the Examiner would consider the phrased "disposed in said markup" to be ambiguous or not enabled. The phrase "said markup" is readily discernable. Moreover, the modified "disposed in" merely describes that the object at issue (e.g., a library reference) is found within (i.e., disposed in) the markup.

Validation Shell Function

This term is found, for example, in each of claims 4 and 5, which recite "a validation shell function encapsulating said function call." With regard to this phrase, reference is made to paragraph [0023] of Appellant's disclosure, which is reproduced below:

> In further illustration of the foregoing process, FIG. 3 is a flow chart illustrating a method for lightweight, client side pattern validation. Beginning in block 310, input data can be accepted in a markup language defined form. In decision block 320, it can be determined whether the input data is to be submitted for processing. Once the data is determined to be submitted for processing, in block 330 the input data provided in selected ones of the input fields can be passed to a validation shell. The validation shell can be a single function defined within the markup language document in which multiple pattern validation operations for corresponding multiple input fields can be processed within a single function call. (emphasis added)

Additional discussion can also be found in paragraphs [0024] and [0010]. Thus, not only is the term "validation shell function" enabled, one having ordinary skill in the art would have no difficultly understanding the scope of this term, particularly when reasonably interpreted in light of the written description of the specification.

Pattern Validation Routine

With regard to this phrase, the Examiner has argued the following on page 8 and also in the paragraphs spanning pages 8 and 9 of the Third Office Action:

> The term "pattern validation routine" is not defined in the specification. Upon examination of the claims and the specification, it is the Examiner's belief that the Applicant intended the term to be the comparison of input with valid input patterns within the validation routine, and will be so read for the remainder of this Office Action.

Appellant is entirely unclear why the Examiner believes this phrase is both not enabled by Appellant's specification and indefinite. The term, on its face, is definite. Moreover, this term is enabled since this term is found in Appellant's originally filed claim 10, which recites:

> defining a pattern validation routine to validate form based input provided through a prototype interface to said routine based upon a validation pattern also provided through said prototype interface.

The concept of "pattern validation" is found throughout Appellant's disclosure and is even found within the Title of the Invention. The phrase "pattern validation routine," as recited in claim 10, is a routine (i.e., method/process) "to validate form based input provided through a prototype interface to said routine based upon a validation pattern also provided through said prototype interface." Thus, the term at issue can easily be defined as a process of pattern validation.

Pervasive Device

In the second amendment, Appellant introduced new claim 15, which recites that "the client device is a pervasive device." With regard to the definition of "pervasive device," reference is made to the "Background of the Invention" section of U.S. Patent No. 6,925,481, which states the following:

> Pervasive devices (also referred to as "pervasive computing devices") have become popular in recent years as people increasingly seek "anywhere, anytime" access to services such as voice and data communications. Many pervasive devices are designed to be mobile, and may equivalently be referred to as "mobile devices" or "mobile computing devices". Examples of mobile pervasive devices range from two-way pagers to personal digital assistants, or "PDAs" (such as the Palm Pilot, Handspring Visor.TM., or Compaq iPAQ) to cellular phones (such as the Nokia 6110) to multi-function devices (such as the Nokia 9110 or Qualcomm "pdQ.TM." smartphone). ("Visor" is a trademark of Handspring, and "pdQ" is a trademark of QUALCOMM Incorporated.) All pervasive devices are not necessarily mobile, however. Examples of this latter category include smart appliances for the home or business setting, devices which are permanently mounted in automobiles, and so forth.
>
> Pervasive devices typically share several common characteristics:
>
> 1) limited processor speed;
>
> 2) limited memory capacity;
>
> 3) small size, which limits the richness of the data input and output interfaces (for example, small screen, limited keypad, and so forth);
>
> 4) a limited amount of software pre-installed on the device; and
>
> 5) access to limited-bandwidth networks.

Reference is also made to paragraphs [0006], [0007], and [0017] of Appellant's disclosure, which refers to an embodiment of the client device being a mobile device. Thus, not only is the term "pervasive device" enabled, one having ordinary skill in the art would have no difficultly understanding the scope of this term, particularly when reasonably interpreted in light of the written description of the specification.

Validation Script Library

With regard to this phrase, the Examiner has argued the following on pages 6 and 9 of the Third Office Action:

> The term "validation script library" is defined in the specification by its use only, which is read by the Examiner, based on a review of the claims and specification, as a client side input validator, and will be so read for the remainder of this Office Action.

In response to a similar argument, Appellant argued the following on pages 7 and 8 of the Second Amendment. Independent claims 1 and 10 recite a "validation script library" that packages a validation process/routine and independent claims 6 and 11 recite a "validation process" disposed in a "validation library." On page 3 of the Second Office Action, the Examiner asserted that the term "validation script library" is construed as a "client side input validator." Appellant questions, however, how the Examiner arrived at this interpretation. Specifically, whereas the term "validation script library" implies that the validation script is located within a "library," the Examiner's interpretation of "client side input validator" implies that the input validator (i.e., corresponding to the claimed validations script) is located in the client.

Notwithstanding that Appellant has clarified the invention recited in claim 1 by reciting that the validations script device is within the client device, the Examiner's interpretation of "validation script library" improperly broadens the scope of the claimed term beyond the reasonable broadest interpretation of the term by one having ordinary skill in the art. The

Examiner's analysis provides little explanation as to why the Examiner believes "library" and "client side" to be comparable. By analogy, the Examiner's analysis would interpret the phrase "a computer disposed within a library," which happens to be within a building, to mean "a computer is disposed within the building." In essence, the Examiner has completely ignored the limitation of "library."

Appellant also notes that a discussion of the claimed library is found throughout Appellant's disclosure, for example, in paragraphs [0018], [0019], [0021], and [0022]. In particular, reference is made to paragraph [0021], which is reproduced below:

> In this regard, a reference 210 to a lightweight script validation library can be disposed within the markup language document 200. The lightweight script validation library itself can include a validation process which can assert compliance between a provided value and a provided pattern. In this way, the library can remain generic and flexible, yet small in size and resource requirements such that the library can be deployed in resource limited devices. Most importantly, the size of the library need not change regardless of the number of patterns provided against which values in the form 230 are to be validated.

Thus, not only is the term "validation script library" enabled, one having ordinary skill in the art would have no difficultly understanding the scope of this term, particularly when reasonably interpreted in light of the written description of the specification.

THE REJECTION OF CLAIMS 1-15 UNDER 35 U.S.C. § 102 FOR ANTICIPATION BASED UPON SCHOLZ

For convenience of the Honorable Board in addressing the rejections, claims 1-3, 10 and 15 each stand or fall alone. Claim 5 stands or falls together with dependent claim 4. Claims 7-9 and 11-14 stands or falls together with independent claim 6.

The Examiner has rejected the claims based upon Scholz in each of the First, Second, and Third Office Actions. For ease of reference, Appellant will reproduce Appellant's prior arguments since these arguments still apply to the present rejection.

Appellant's Arguments in First Amendment

The factual determination of anticipation under 35 U.S.C. § 102 requires the identical disclosure, either explicitly or inherently, of each element of a claimed invention in a single reference.[1] As part of this analysis, the Examiner must (a) identify the elements of the claims, (b) determine the meaning of the elements in light of the specification and prosecution history, and (c) identify corresponding elements disclosed in the allegedly anticipating reference.[2] This burden has not been met. Moreover, the Examiner has failed to clearly designate the teachings in Scholz being relied upon the statement of the rejection. In this regard, the Examiner's rejection under 35 U.S.C. § 102 also fails to comply with 37 C.F.R. § 1.104(c).[3]

Despite this requirement, the Examiner's statement of the rejection simply consists of the Examiner repeating, almost word-for-word, each of the recited claims and asserting that the entire claim is disclosed by certain specified passages within Scholz. The manner in which the Examiner conveyed the statement of the rejection, however, has not "designated as nearly as practicable" the particular parts in Scholz being relied upon in the rejection.

[1] In re Rijckaert, 9 F.3d 1531, 28 USPQ2d 1955 (Fed. Cir. 1993); Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick Co., 730 F.2d 1452, 221 USPQ 481 (Fed. Cir. 1984).

[2] Lindermann Maschinenfabrik GMBH v. American Hoist & Derrick Co., supra.

[3] 37 C.F.R. § 1.104(c) provides:

> In rejecting claims for want of novelty or for obviousness, the examiner must cite the best references at his or her command. When a reference is complex or shows or describes inventions other than that claimed by the applicant, the particular part relied on must be designated as nearly as practicable. The pertinence of each reference, if not apparent, must be clearly explained and each rejected claim specified.

It is practicable for the Examiner, for each of the claimed elements, to specifically identify each feature within Scholz being relied upon to teach each of the particular claimed elements. For example, the Examiner can "specifically identify" a feature, corresponding to the claimed element, within the applied prior art by identifying a reference numeral associated with the feature. In addition to, or alternatively, the Examiner may cite to a brief passage (i.e., 1 or 2 lines or even a portion of a line) within the applied prior art that identifies the feature that corresponds to the claimed element. However, merely citing a long passage or an entire paragraph to disclose a single (or multiple) claimed elements does not designate "as nearly as practicable," the particular features within Scholz being relied upon by the Examiner in the rejection.

Appellant also notes the Examiner reference to M.P.E.P. § 2143 in the paragraph spanning page 9 and 10 of the Office Action, in which the Examiner asserted that "any citations to specific, [sic] pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way." This assertion, however, is not supported by M.P.E.P. § 2143, entitled "Rejection Over Prior Art's Broad Disclosure Instead of Preferred Embodiments." Moreover, notwithstanding that "patents are relevant as prior art for all they contain," the Examiner must still meet the requirements of 37 C.F.R. § 1.104(c).

The importance of the specificity requirement of 37 C.F.R. § 1.104(c) is evident in M.P.E.P. § 706.07, which states:

> The examiner should never lose sight of the fact that in every case the applicant is entitled to a full and fair hearing, and that a clear issue between applicant and examiner should be developed, if possible, before appeal.

A clear issue, however, cannot be developed between Appellant and the Examiner where the basis for the Examiner's rejection of the claims is ambiguous. The Examiner's "analysis" provides little insight as to (i) how the Examiner is interpreting the elements of the claims and (ii) what specific features within Scholz the Examiner believes identically discloses the specific elements (and interactions between elements) recited in the claims. By failing to specifically identify those features within Scholz being relied upon in the rejection, the Examiner has essentially forced Appellant to engage in mind reading and/or guessing to determine how the Examiner is interpreting the elements of the claims and what specific features within Scholz the Examiner believes identically disclose the claimed invention. Appellant also notes that any continuing disagreement between Appellant and the Examiner as to whether or not a particular claimed feature is disclosed by Scholz is a direct result of a lack of specificity by the Examiner in the statement of the rejection.

Claim 1

Independent claim 1 recites the following features:

> validation processor configured with a prototype interface for receiving both a field validation pattern and also form based input to be validated against said field validation pattern; and,
>
> a validation script library packaging said validation processor

To teach these limitations the Examiner cited paragraphs [0116] and [0131] of Scholz and asserted that "a 'form processor' as a separate component or module" and "the custom tags, for the validation, implemented as an object model stored in the tag library."

For ease of reference, paragraphs [0116] and [0131] of Scholz are reproduced below:

> [0116] Alternatively, a non-Java-oriented programming technique may be used as the form processor 808. For example, the form processor 808 may be a separate component or module (e.g., software, firmware, and/or hardware) that analyzes the form definitions 806 to identify the custom tags. These custom tags are then replaced with the corresponding HTML code, validation code, and optionally a call to the corresponding validation code.
> [0131] In one implementation, the custom tags are implemented as an object model (e.g., stored in the tag library 816). An exemplary object model to be used by the form tags to store attributes for each input type and the overall form is illustrated in the following tables. An initial object is the FormCollection object, illustrated in Table 16:

The features recited in claim 1 include: (i) a validation processor; (ii) a prototype interface; (iii) a field validation pattern; (iv) form based input; (v) a validation script library; and (vi) all the interactions between features (i)-(v). The Examiner, however, only asserted that Scholz discloses a form processor 808 and custom tags stored in a tag library 816.

Moreover, the Examiner's rejection is ambiguous as to what particular features in Scholz allegedly disclose the particular features recited in the claims. Assuming, for sake of argument, that the Examiner is asserting that the form processor 808 identically discloses the claimed validation processor and the tag library 816 identically discloses the claimed validation script library, then these features fail to identically disclose the claimed invention. As illustrated in Fig. 8 of Scholz the tag library 816 is not "packaging" the form processor, as recited in claim 1. Instead, the tag library 816 is separate from the form processor 808. Moreover, the form processor 808 does not <u>receive</u> "form based input" (i.e., input received in a form), as recited in claim 1. Instead, the form processor 808 receives input form definitions 806 from which the form processor 808 creates output form definitions 818 (see Fig. 8 of Scholz).

The Examiner has also failed to establish that Scholz teaches "a prototype interface" within the validation processor that receives "a field validation pattern." Therefore, for all the reasons stated above, Appellant submits that Scholz fails to identically disclose the claimed invention, as recited in claim 1, within the meaning of 35 U.S.C. § 102.

Claim 2

Claim 2 recites the following features: (i) a library reference; (ii) markup; (iii) a form; (iv) at least one form based input field programmed for validation using said validation processor; (v) a function call; (vi) a configuration for passage a reference; (vii) a value; and (viii) all the interactions between these features and also the features recited in claim 1. However, to teach the limitations recited in claim 2, the Examiner only cited paragraph [0109] to teach "validation by reference to the validation code" and paragraphs [0131]-[0132] to teach "custom tag library and the FormCollection object." Similar to the rejection of claim 1, the Examiner's statement of rejection is ambiguous as to what particular features in Scholz allegedly disclose the particular features recited in the claims. Moreover, it is readily apparent that the Examiner has not established that Scholz identically discloses each of the limitations recited in claim 2 because there are more limitations recited in claim 2 than the features mentioned by the Examiner in the statement of the rejection.

Notwithstanding the Examiner's continued lack of specificity and comprehensiveness in alleging that the claimed limitations are identically disclosed in Scholz, it is readily apparent that even those features identified by the Examiner do not identically disclose the limitations recited in the claims. For ease of reference, paragraph [0019] is reproduced below:

> [0119] Custom Form Tag: The custom form tag extends the HTML form tag by 10 providing automated form validation creation. This tag also supports existing HTML form tag attributes. The custom form tag is illustrated in Table 4.

The above-cited paragraph refers to custom tags, which Scholz has identified with reference numeral 802 (3rd sentence in paragraph [0104]). Notwithstanding the custom tags are replaced by executable code from the tag library 816 (see paragraphs [0106]-[0107]), the markup (i.e., the input form definitions 806) in which the custom tags 802 are found (see paragraph [0104]), does not "[define] a form having at least one form based input field programmed for validation," as recited in claim 2.

Regarding the Examiner's citation of paragraphs [0131]-[0132] to teach "custom tag library and the FormCollection object," these features do not identically disclose "a function call to said validation processor further disposed in said markup, said function call having a configuration for passing a reference to a value in said at least one form based input field for validation in said validation processor," as recited in claim 2. As illustrated in Fig. 8 of Scholz, the markup (i.e., the input form definitions 806) does not include a function call to the validation processor (i.e., form processor 808). Therefore, for all the reasons stated above, Appellant submits that Scholz fails to identically disclose the claimed invention, as recited in claim 2, within the meaning of 35 U.S.C. § 102.

Claim 3

Similar to claim 2, the markup (i.e., the input form definitions 806) does not include function calls to the validation processor (i.e., form processor 808). Therefore, Appellant submits that Scholz fails to identically disclose the claimed invention, as recited in claim 2, within the meaning of 35 U.S.C. § 102.

Claims 4 and 5

Since Scholz fails to identically disclose the claimed function call, Scholz cannot disclose a validation shell function encapsulating the function call.

Claims 6 and 11

Regarding independent claims 6 and 11, the Examiner generally cited paragraphs [0092]-[0175][4] to teach "retrieving input, passing the input value, and validating the retrieved value according to a valuation pattern within the content browser. The Examiner specifically cited paragraph [0092] as teaching "the validation with a markup language, HTML and/or XML, in a client-side valuation.). For ease of reference, paragraph [0092] is reproduced below:

> [0092] Users are able to input requests to an application via a user interface that presents one or more forms to the user, each form having one or more data input fields (e.g., text areas, user-selectable check boxes or buttons, etc.). These data inputs are predominately referred to herein as user inputs, although the inputs can alternatively come from elsewhere (e.g., from another application or component). For many forms, the application developer desires to place restrictions on the data that can be input to the fields of the form. An automatic input validation technique is used that allows forms with input fields to be automatically generated to include input validation for one or more of the input fields. Forms can be automatically generated in any of a wide variety of languages, and in one embodiment are generated as conventional pages (documents) of a conventional markup language such as the well-known HyperText Markup Language (HTML) or the well-know extensible Markup Language (XML). The form itself includes the validation code and thus performs the validation at the client (referred to as client-side validation).

[4] Paragraphs [0092]-[0175] constitute pages 11-23 of Scholz, and is another example of the Examiner failing to designate, as nearly as practicable, the particular parts in Scholz being relied upon in the rejection

One of the key differences between the claimed invention, as recited in claims 6 and 11 (or any of the other claims), and the teachings of Scholz is found in the last sentence of paragraph [0092], which states that "[t]he form itself includes the validation code and thus performs the validation at the client (referred to as client-side validation)."

As illustrated in Fig. 2 of Appellant's disclosure and recited in the claims, the script validation library 210, which includes the validation process, is separate from the form 230. For example, claims 6 and 11 recite that "a value ... from a form" is retrieved and the "retrieved value along with a validation pattern" is passed "to a validation process disposed within a lightweight validation library." Thus, the claims recite that the validation process is separate from the form, whereas Scholz teaches that the form itself performs the validation. Therefore, Appellant submits that Scholz fails to identically disclose the claimed invention, as recited in claims 6 and 11, within the meaning of 35 U.S.C. § 102.

Claim 10

Regarding independent claim 10, the Examiner identically repeated, word-for-word, the statement of the rejection presented with regard to independent claim 6. Appellant notes, however, that claim 10 is substantially different the claim 6, yet this difference has not been reflected in the Examiner's statement of the rejection. Notwithstanding this ambiguous and incomplete rejection, Appellant incorporates herein the arguments previously presented with regard to claim 6 as also applying to claim 10. Claim 10 recites that the validation process is separate from at least one form based input field in markup, whereas Scholz teaches that the form/markup itself performs the validation. Thus, Appellant submits that Scholz fails to identically disclose the claimed invention, as recited in claim 10, within the meaning of 35 U.S.C. § 102.

Appellant's Arguments in Second Amendment

Appellant does note that the Examiner has attempted to construe certain claim limitations. However, in certain instances, the Examiner's analysis fails to properly establish the ordinary and customary meaning associated with these claims limitations given their broadest reasonable interpretation by one having ordinary skill in the art.

For example, independent claims 1 and 10 recite a "validation script library" that packages a validation process/routine and independent claims 6 and 11 recite a "validation process" disposed in a "validation library." On page 3 of the Final Office Action, the Examiner that the term "validation script library" is construed as a "client side input validator." Appellant questions, however, how the Examiner arrived at this interpretation. Specifically, whereas the term "validation script library" implies that the validation script is located within a "library," the Examiner's interpretation of "client side input validator" implies that the input validator (i.e., corresponding to the claimed validations script) is located in the client.

Notwithstanding that Appellant has clarified the invention recited in claim 1 by reciting that the validations script device is within the client device, the Examiner's interpretation of "validation script library" improperly broadens the scope of the claimed term beyond the reasonable broadest interpretation of the term by one having ordinary skill in the art. The Examiner's analysis provides little explanation as to why the Examiner believes "library" and "client side" to be comparable. By analogy, the Examiner's analysis would interpret the phrase "a computer disposed within a library," which happens to be within a building, to mean "a computer is disposed within the building." In essence, the Examiner has completely ignored the limitation of "library."

As another example, the Examiner asserted that the term "disposed within markup" means that "the invention used a markup computer language." How the Examiner arrived at this interpretation, however, is again unclear. Notwithstanding the Examiner's lack of analysis, the Examiner has improperly failed to recognize that the term "disposed within markup" establishes a relationship between a library reference and markup such that "a library reference ... [is] disposed within markup," as recited in claim 2. By merely asserting that the invention uses a markup computer language, the Examiner has ignored this claimed relationship between the library reference and markup.

Claim 1

In the prior Amendment, Appellant argued the following with regard to claim 1:

> Moreover, the Examiner's rejection is ambiguous as to what particular features in Scholz allegedly disclose the particular features recited in the claims. Assuming, for sake of argument, that the Examiner is asserting that the form processor 808 identically discloses the claimed validation processor and the tag library 816 identically discloses the claimed validation script library, then these features fail to identically disclose the claimed invention. As illustrated in Fig. 8 of Scholz the tag library 816 is not "packaging" the form processor, as recited in claim 1. Instead, the tag library 816 is separate from the form processor 808. Moreover, the form processor 808 does not receive "form based input" (i.e., input received in a form), as recited in claim 1. Instead, the form processor 808 receives input form definitions 806 from which the form processor 808 creates output form definitions 818 (see Fig. 8 of Scholz).

The Examiner's response to Appellant's arguments regarding claim 1 is found on page 14 of the Final Office Action. The Examiner, however, did not address the substance of these arguments. Specifically, Appellant based these arguments based upon certain interpretations of Scholz with regard to the claimed limitations. The Examiner, however, neither corrected Appellant's interpretations as to certain teachings in Scholz allegedly disclosing certain claimed features nor refuted Appellant's accompanying analysis. In this regard, the Examiner is referred to M.P.E.P. § 707.07(f), which states that "the Examiner, if he or she repeats the rejection, take note of the applicant's argument and answer the substance of it." The Examiner's response on page 14 is merely a blanket statement that the Examiner disagrees with Appellant without any accompanying analysis.

Moreover, Appellant notes that claim 1 has been amended to clarify that "the validation processor separate from said markup." In contrast, paragraph [0092] of Scholz specifically teaches the opposite of this limitation when Scholz teaches that "[t]he form itself includes the validation code and thus performs the validation at the client." Thus, Scholz fails to identically disclose the claimed invention, as recited in claim 1, within the meaning of 35 U.S.C. § 102.

Claims 6 and 11

With regard to independent claims 6 and 11, Appellant previously argued that "the claims recite that the validation process is separate from the form, whereas Scholz teaches that the form itself performs the validation." The Examiner's response to this argument is found on pages 16 and 17 of the Final Office Action, in which the Examiner asserted:

> Applicant argues that Scholz does not teach a separate form validation process.
> The Examiner disagrees.
> See, Scholz, paragraph [0116], teaching a "form processor" as a separate component or module.

Despite Appellant referring to this passage in the prior Amendment and also above in the present Amendment, the Examiner is again referred to the last sentence of paragraph [0092] of Scholz, which unambiguously states that "[t]he form itself includes the validation code and thus performs the validation at the client (referred to as client-side validation)."

The Examiner's reference to the form processor 808 is inapposite. Paragraph [0105] of Scholz describes that the "form processor 808 generates a temporary form definition 814" and that the purpose of the form processor 808 illustrated in Fig. 8 of Scholz is to generate "output form definitions 818," which are "written in a source code that defines the contents of the forms" (see paragraph [0113]). Thus, the form processor 808 is not comparable to the claimed validation process. Instead, the form processor 808 outputs validation code that is used within a form to validate form based input. Thus, Appellant submits that Scholz fails to identically disclose the claimed invention, as recited in claims 6 and 11, within the meaning of 35 U.S.C. § 102.

Claim 10

Claim 10 recites that the validation process is separate from at least one form based input field in markup, whereas Scholz teaches that the form/markup itself performs the validation. Thus, Appellant submits that Scholz fails to identically disclose the claimed invention, as recited in claim 10, within the meaning of 35 U.S.C. § 102.

Examiner's Response in the Third Office Action

The Examiner's response to Appellant's prior arguments is found on pages 23-28 of the Third Office Action.

On pages 23 and 24, the Examiner disagreed with Appellant's arguments as to the Examiner's improperly interpreting the phrase "validation script library." In particular, the Examiner asserted the following:

> **FIRST**: Applicant argues that the Examiner's interpretation of the term "validation script library" in the Final Office Action was in error. See, Remarks, pages 7-8. Specifically, Applicant argues that the interpretation of the terms as a "client side input validator" is without evidentiary basis.
>
> The Examiner disagrees.
>
> In the Final Office Action, the Examiner noted: "The term 'validation script library' is defined in the specification by its use only, which is read by the Examiner, based on a review of the claims and specification, as a <u>client side input validator</u> ... " See, Final Office Action, page 3, emphasis added.
>
> The specification defines a "validation script library" in the following contexts:
>
> a) "The system further can include a <u>validation script library packaging the validation process</u>." See, disclosure, paragraph [0008], emphasis added. This is the evidence that the term, in its broadest reasonable interpretation, was a "validator."
>
> b) "Moreover, as the validation processor 150 can be packaged within a lightweight validation script library 140, there will be no need to cache a complete copy of a conventional script library thus permitting the lightweight validation script library 140 to remain <u>resident in the client computing device</u> 130 -- even where the resources of the client computing device 130 are limited." See, disclosure, paragraph [0019], emphasis added. This is the evidence that the term, in its broadest reasonable interpretation, was a "client side" device. See also, figure 1, showing the "lightweight validation script library" 140 residing on the client device 130.
>
> c) See, figures 1-3, showing the validation process as part of the input. Particularly see, figure 3, element 330, as the step to "pass input data to validation shell." This is the evidence that the "validation script library received input.
>
> Therefore, in its broadest reasonable interpretation, the term "validation script library" may reasonably be interpreted as a "client side input validator." (emphasis in original)

The Examiner appears to ignore the distinction between <u>including</u> a validator and <u>being</u> a validator. Moreover, even if the validation script library is interpreted as being a validator, as asserted by the Examiner, the Examiner has failed to establish how this term necessarily encompasses the limitation of "client-side," as alleged by the Examiner. In fact, Applicant has specifically claimed that the "validation script library [is] within said client device." In so doing, Appellant has clearly implied that a validation script library, in general, does not necessary have to be within a client device. However, for the purpose of claim 1, for example, the validation script library <u>is</u> claimed to be within the client device.

The Examiner's assertion that "[t]his is the evidence that the term, in its broadest reasonable interpretation, was a 'client side' device" is akin to asserting that the phrase "vending machine" in its broadest reasonable interpretation is an office vending machine because this hypothetical vending machine was described in the hypothetical specification as being within an office. Notwithstanding the reason why one would want a vending machine in an office, just because a specification discloses that the vending machine is in the office does not necessarily allow for phrase "vending machine" to be reasonably interpreted as an "office vending machine."

On page 24, the Examiner presented the following arguments:

> **SECOND**: Applicant further argues: "the term 'validation script library' implies that the validation script is located within a 'library'." See, Remarks, pages 7-8.
>
> The Examiner disagrees.
>
> The term "validation script library" implies that the validation script is a library, not that it resides within one. There is not support found in the original claims or specification that a "validation script" is separate from the "library."

Consider the phrases "medical library," "children's library," and "music library." One having ordinary skill in the art of libraries would recognize these terms respective refer to a library of medical books, a library of children's book, and a library of music (e.g., CD, LPs, etc.). This is comparable to Appellant's claim construction of the phrase "validation script library" as a library of validation script. The Examiner's claim construction, however, has not been established as being reasonable or factually supported. On the contrary, the Examiner claim construction does not appear to make logical sense.

On page 26, with regard to the phrase "disposed in markup," the Examiner asserted that this phrase "was not known to have a definition known to one of ordinary skill in the art at the time of the invention to the knowledge of the Examiner." As already argued the above, the Examiner's analysis in unsound. The Examiner further asserted on page 27 the following:

> Applicant states without reference or further explanation that the term "disposed within markup" establishes a relationship between a library reference and markup such that a library reference is disposed within markup. This is merely a circuitous statement without foundation in the specification. Finally, there is no definition or differentiation of "markup" found in Applicant's argument that would distinguish the term over the customary and ordinary usage of the term, known to one of ordinary skill in the art at the time of the invention, as identifying a markup language.

This statement by the Examiner is absolutely incorrect since claim 2 clearly "establishes a relationship between a library reference and markup such that a library reference is disposed within markup." Specifically, claim 2 recites:

> a library reference to said script library disposed in said markup; and,
>
> a function call to said validation processor further disposed in said markup, said function call having a configuration for passing a reference to a value in said at least one form based input field for validation in said validation processor. (emphasis added)

Reference is also made to paragraph [0009] of Appellant's disclosure, which states "a library reference to the script library can be disposed within markup defining a form." Reference is also made to paragraph [0021] of Appellant's disclosure, which states "reference 210 to a lightweight script validation library can be disposed within the markup language document 200." Thus, the library reference is disposed (i.e., located) within markup.

Based upon the above interpretation with regard to the phrase "disposed in said markup," the Examiner's assertion on page 27 that "in its broadest reasonable interpretation, the term 'disposed within markup' is read as being written in a markup language," is grossly incomplete. Thus, the Examiner has failed to properly construe this term, and as a result, has ignored limitations recited in the claims.

On pages 27 and 28, the Examiner further asserted the following:

> Applicant argues that "Scholz fails to teach "the validation processor separate from said markup." Applicant further argues that Scholz teaches the opposite of the limitation of a processor separate from a markup. See, Remarks, pages 9-11.
>
> The Examiner disagrees.
>
> Initially, it is noted that the term "markup" is not defined in the specification, as noted in several rejections above. The broadest reasonable interpretation is that "markup" means some form of markup language.
>
> Scholz clearly teaches that a validation processor may be separate from the form or its markup language expression. See, Scholz, paragraph [0014], teaching that the code is within a validation processor receiving input from the form. The processor, residing on the client, is separate from the code in the form that receives the data and processed on the processor. The processor receiving the function calls to process the data is separate from the input of the data on the form.

At the outset, Appellant notes that the Examiner's cited paragraph of [0014] appears to be in error since this paragraph only refers to the description of Fig. 9 of Scholz. Notwithstanding the ambiguity of what teachings within Scholz the Examiner is relying upon, the Examiner's assertions ignores the plain teachings of Scholz. Specifically, paragraph [0092] of Scholz <u>specifically teaches the opposite</u> of this limitation when Scholz teaches that "[t]he form itself includes the validation code and thus performs the validation at the client." Thus, whereas the claims recite that the validation processor is separate from markup (which defines the form), Scholz teaches that the form (and thus the markup language that defines the form, as described in lines 13-18 of paragraph [0092]) includes the validation processor.

Although Appellant has addressed these issues previously, Appellant again notes the Examiner failure to set forth a proper rejection per the requirements of 37 C.F.R. § 1.104(c). For example, on pages 12 and 13 of the Third Office Action with regard to claim 1, the Examiner improperly <u>generalizes</u> the claimed invention, while generally referring to paragraphs [0092]-[0175] of Scholz to teach the claimed invention.

Therefore, for the reasons stated above, Appellant respectfully submits that the imposed rejection of claims 1-15 under 35 U.S.C. § 102 for anticipation based upon Scholz is not viable.

Conclusion

Based upon the foregoing, Appellant respectfully submits that the Examiner's rejections under 35 U.S.C. §§ 112, 102 are not viable. Appellant, therefore, respectfully solicits the Honorable Board to reverse the Examiner's rejection under 35 U.S.C. §§ 112, 102.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due under 37 C.F.R. §§ 1.17, 41.20, and in connection with the filing of this paper, including extension of time fees, to Deposit Account 09-0461, and please credit any excess fees to such deposit account.

Date: April 16, 2007

Respectfully submitted,

/Scott D. Paul/
Scott D. Paul
Registration No. 42,984
Steven M. Greenberg
Registration No. 44,725
Phone No. 561-922-3845
CUSTOMER NUMBER 46320

VIII. CLAIMS APPENDIX

1. A lightweight pattern validation system for a client device receiving markup defining a form, comprising:

a validation processor separate from said markup and configured with a prototype interface for receiving both a field validation pattern and also form based input to be validated against said field validation pattern; and,

a validation script library within said client device and packaging said validation processor, wherein

the form has at least one form based input field programmed for validation using said validation processor.

2. The system of claim 1, further comprising:

a library reference to said script library disposed in said markup; and,

a function call to said validation processor further disposed in said markup, said function call having a configuration for passing a reference to a value in said at least one form based input field for validation in said validation processor.

3. The system of claim 2, further comprising

a plurality of additional function calls to said validation processor disposed in said markup, each additional one of said functional calls having a configuration for passing a reference to a value in a corresponding form based input field for validation in said validation processor.

4. The system of claim 2, further comprising a validation shell function encapsulating said function call.

5. The system of claim 3, further comprising a validation shell function encapsulating said function call.

6. A pattern validation method comprising the steps of:

retrieving a value for a form based input field from a form defined in markup rendered in a content browser;

passing said retrieved value along with a validation pattern for said form based input field to a validation process disposed within a lightweight validation library separate from and coupled to said rendered markup; and,

validating said retrieved value according to said validation pattern in said content browser.

7. The method of claim 6, further comprising the step of repeating said retrieving, passing and validating steps for at least one additional value for at least one additional form based input field disposed in said markup rendered in said content browser.

8. The method of claim 6, further comprising the step of performing said retrieving, passing, and validating steps in a validation shell function disposed in said markup rendered in said content browser.

9. The method of claim 7, further comprising the step of performing said retrieving, passing, validating and repeating steps in a validation shell function disposed in said markup rendered in said content browser.

10. A pattern validation method comprising the steps of:

defining a pattern validation routine to validate form based input provided through a prototype interface to said routine based upon a validation pattern also provided through said prototype interface;

packaging said pattern validation routine into a lightweight validation script library;

referencing said lightweight validation script library in markup disposed within a content server configured to distribute said markup to requesting clients;

defining at least one form based input field in said markup and further defining a validation pattern for each of said at least one form based input fields; and,

for each form based input field and defined validation pattern, disposing a function call to said pattern validation routine in said lightweight script library.

11. A machine readable storage having stored thereon a computer program for pattern validation, the computer program comprising a routine set of instructions which when executed by the machine cause the machine to perform the steps of:

retrieving a value for a form based input field from a form defined in markup rendered in a content browser;

passing said retrieved value along with a validation pattern for said form based input field to a validation process disposed within a lightweight validation library separate from and coupled to said rendered markup; and,

validating said retrieved value according to said validation pattern in said content browser.

12. The machine readable storage of claim 11, further comprising the step of repeating said retrieving, passing and validating steps for at least one additional value for at least one additional form based input field disposed in said markup rendered in said content browser.

13. The machine readable storage of claim 11, further comprising the step of performing said retrieving, passing, and validating steps in a validation shell function disposed in said markup rendered in said content browser.

14. The machine readable storage of claim 12, further comprising the step of performing said retrieving, passing, validating and repeating steps in a validation shell function disposed in said markup rendered in said content browser.

15. The system of claim 1, wherein the client device is a pervasive device.

IX. EVIDENCE APPENDIX

No evidence submitted pursuant to 37 C.F.R. §§ 1.130, 1.131, or 1.132 of this title or of any other evidence entered by the Examiner has been relied upon by Appellant in this Appeal, and thus no evidence is attached hereto.

X. RELATED PROCEEDINGS APPENDIX

Since Appellant is unaware of any related appeals and interferences, no decision rendered by a court or the Board is attached hereto.